ธาวินทร์ ลีลาคุณารักษ์

SOFT MAIL

บาร์โค้ดชนิด Postnet

บาร์โค้ดแบบ Postnet ที่ใช้แทนค่ารหัสไปรษณีย์ โดยแทนค่ารหัสไปรษณีย์ที่ใช้อยู่ในแต่ละประเทศ

ผมไม่ได้นำเรื่องเกี่ยวกับบาร์โค้ดมาเสนอนานพอสมควร สาเหตุเนื่องจากพักหลัง ๆ ผมไม่ค่อยมีเวลาที่จะค้นคว้าหาข้อมูลต่าง ๆ มากนัก แต่ก็ได้ฝากเพื่อน ๆ หรือน้อง ๆ ให้ช่วยหาข้อมูลเกี่ยวกับบาร์โค้ดมาให้ ซึ่งก็ได้เพิ่มเติมมาเป็นระยะ ๆ และหากมีเวลา ผมจะพยายามนำเรื่องราวมาเสนอเป็นลำดับ ๆ ไป สำหรับเรื่องราวของบาร์โค้ดที่ผมนำมาเสนอในครั้งนี้ จะเป็นเรื่องของบาร์โค้ดที่เรียกกันทั่วไปว่า "บาร์โค้ดแบบ Postnet" ซึ่งเป็นบาร์โค้ดที่ใช้แทนค่ารหัสไปรษณีย์ (ZIP CODE) ที่ใช้ในสหรัฐอเมริกา และประเทศในแถบยุโรป แต่ในบ้านเรายังไม่มีการนำมาใช้

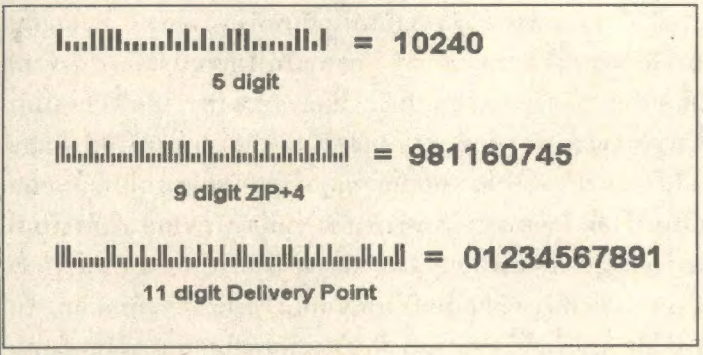
บาร์โค้ดแบบ Postnet จัดเป็นบาร์โค้ดที่ใช้แทนค่าหรือข้อมูลที่เป็นตัวเลขชนิดหนึ่ง ที่มีระบบตรวจสอบความถูกต้องของรหัส (CHECK DIGIT) อยู่ในตัวเอง ซึ่งสามารถใช้แทนตัวเลขตั้งแต่ 0 ถึง 9 เช่นเดียวกับบาร์โค้ดแบบ UPC, EAN หรือ Interleave 2 of 5 ที่ผมเคยนำเรื่องราวมาเสนอในช่วงก่อนหน้านี้ บาร์โค้ดแบบ Postnet นี้จะเป็นบาร์โค้ดที่ใช้พิมพ์ลงบนหน้าซองจดหมายหรือพัสดุไปรษณีย์ โดยจะใช้แทนค่ารหัสไปรษณีย์ที่ใช้อยู่ในแต่ละประเทศ

โดยปกติบาร์โค้ดแบบ Postnet จะเป็นบาร์โค้ดที่สามารถแทนข้อมูลที่เป็นตัวเลขได้ไม่จำกัดจำนวนหลัก แต่โดยทั่วไปแล้ว จะยึดจำนวนหลักของรหัสไปรษณีย์ที่ใช้ในแต่ละประเทศเป็นเกณฑ์ ซึ่งในบางประเทศอาจจะใช้รหัสไปรษณีย์แบบ 5 หลัก 9 หลัก (ZIP+4) หรือ 11 หลัก แล้วแต่ระบบของแต่ละประเทศ

รายละเอียดภายในตัวบาร์โค้ด

รูปแบบของแท่งบาร์โค้ด (Bar)

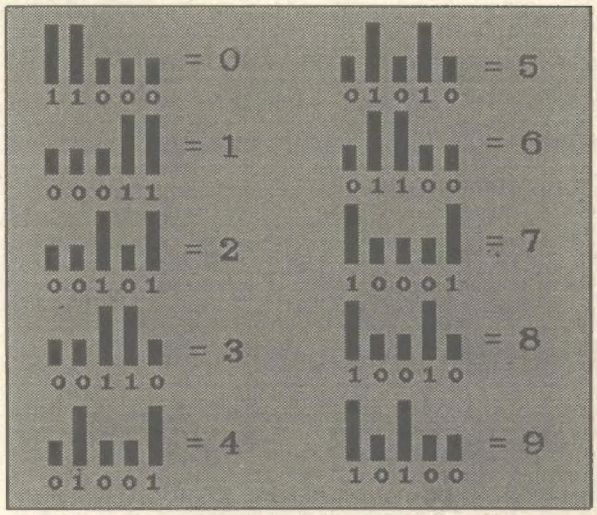
รูปแบบของแถบเส้นบาร์โค้ดที่ใช้ในบาร์โค้ดแบบ Postnet นั้น จะไม่เหมือนกับบาร์โค้ดหลาย ๆ ชนิดที่เราคุ้นเคย (ดูรูปที่ 1) ไม่ว่าจะเป็นบาร์โค้ดในกลุ่ม UPC/EAN ซึ่งจะใช้แท่งสี่เหลี่ยมแบบขาว (สว่าง) กับแท่งสี่เหลี่ยมแบบทึบ (สีเข้ม) ประกอบกันเป็นตัวบาร์โค้ด หรือบาร์โค้ดในกลุ่ม 2 of 5 หรือกลุ่ม Code 3 of 9 (Code 39) ที่ใช้แท่งสี่เหลี่ยมแบบแคบและแท่งสี่เหลี่ยมแบบกว้าง ประกอบกันเป็นตัวบาร์โค้ด แต่ลักษณะของแถบเส้นบาร์โค้ดแบบ Postnet นี้ จะใช้แท่งสี่เหลี่ยมแบบทึบที่มีความสูงต่างกันประกอบกันเป็นตัวบาร์โค้ดแทน โดยที่แท่งสี่เหลี่ยมที่มีความสูงน้อยหรือแท่งสี่เหลี่ยมแบบสั้น จะแทนด้วยค่า 0 ในระบบเลขฐานสอง และแท่งสี่เหลี่ยมที่มีความสูงมากกว่าหรือแท่งสี่เหลี่ยมแบบยาว (ประมาณ 2.5 เท่า ของแท่งสี่เหลี่ยมแบบสั้น) จะใช้แทนค่า 1 ในระบบเลขฐานสอง



รูปที่ 1 บาร์โค้ดแบบ Postnet

แพตเทิร์นของบาร์โค้ดแบบ Postnet

ภายในตัวบาร์โค้ดแบบ Postnet ในส่วนของแถบเส้นบาร์โค้ดที่ใช้แทนข้อมูลที่เป็นตัวเลข (เลขฐานสิบ) จำนวน 1 หลัก จะประกอบด้วยแท่งสี่เหลี่ยมทึบแบบสั้นจำนวน 3 แท่ง และแท่งสี่เหลี่ยมทึบแบบยาวจำนวน 2 แท่งวางสลับกัน ซึ่งชุดของแถบเส้นบาร์โค้ดแบบ Postnet ที่ใช้แทนข้อมูลที่เป็นตัวเลข 1 หลัก จะสามารถแทนด้วยเลขฐานสอง จำนวน 5 หลัก สำหรับรูปแบบ (Pattern) ของบาร์โค้ดที่ใช้ในบาร์โค้ดแบบ Postnet ทั้งหมดจะแสดงอยู่ในรูปที่ 2



รูปที่ 2 รูปแบบของแถบบาร์โค้ดที่ใช้ในบาร์โค้ดแบบ Postnet

ส่วนประกอบของบาร์โค้ดแบบ Postnet

สำหรับส่วนประกอบต่าง ๆ ของบาร์โค้ดแบบ Postnet จะมีอยู่ 4 ส่วน คือ ส่วนเริ่มต้น (start bar) ชุดแถบเส้นบาร์โค้ดที่ใช้แทนรหัสไปรษณีย์ ชุดแถบเส้นบาร์โค้ดที่ใช้แทนค่า CHECK DIGIT และส่วนปิดท้าย (stop bar) ตามลำดับ (ดูรูปที่ 3) โดยรายละเอียดของแต่ละส่วนจะมีดังนี้

รูปที่ 3 ส่วนประกอบของบาร์โค้ดแบบ Postnet

- ส่วนเริ่มต้น (START BAR) และส่วนปิดท้าย (STOP BAR) เป็นส่วนที่ใช้บ่งบอกถึงจุดเริ่มต้นและจุดสิ้นสุดของการอ่านแถบบาร์โค้ดของเครื่องอ่าน (BARCODE SCANNER) โดยที่ส่วนเริ่มต้น (START) และส่วนปิดท้าย (STOP BAR) จะประกอบด้วยแท่งสี่เหลี่ยมทึบแบบยาวจำนวนส่วนละ 1 แท่ง

- ชุดแถบเส้นบาร์โค้ดที่ใช้แทนรหัสไปรษณีย์ เป็นส่วนของชุดแถบเส้นบาร์โค้ดที่ใช้แทนค่าตัวเลขที่ใช้เป็นรหัสไปรษณีย์ ส่วนแถบเส้นบาร์โค้ดในส่วนนี้จะแทนเลขจำนวนกี่หลักนั้น ขึ้นอยู่กับรหัสไปรษณีย์ที่ใช้ สำหรับระยะห่างระหว่างแท่งสี่เหลี่ยมแต่ละแท่ง โดยทั่วไปจะวางห่างกันประมาณเท่ากับความกว้างของแถบเส้นบาร์โค้ดที่ใช้

- ชุดแถบเส้นบาร์โค้ดที่ใช้แทนค่า CHECK DIGIT เป็นชุดของแถบเส้นบาร์โค้ดที่ใช้แทนค่าตัวเลขที่เป็นค่าตรวจสอบความถูกต้องของรหัส ซึ่งเป็นตัวเลขจำนวน 1 หลัก วางต่อจากแถบเส้นบาร์โค้ดที่ใช้แทนรหัสไปรษณีย์

ตัวเลขที่ใช้เป็นค่าตรวจสอบความถูกต้องของรหัส (CHECK DIGIT) ของบาร์โค้ดแบบ Postnet สามารถหาได้โดยนำค่าตัวเลขที่ปรากฏในรหัสไปรษณีย์ทั้งชุดมาบวกกัน แล้วหารด้วย 10 แต่เก็บเฉพาะเศษไว้ จากนั้นให้นำเศษที่ได้ไปลบจาก 10 อีกครั้งหนึ่ง ผลที่ได้จากการลบครั้งสุดท้ายนี้ จะถูกนำไปใช้เป็นค่าตรวจสอบความถูกต้องของรหัสต่อไป (ดูรูปที่ 4)

เนื่องจากบาร์โค้ดแบบ Postnet เป็นบาร์โค้ดที่จะใช้พิมพ์ลงบนซองจดหมายหรือบรรจุภัณฑ์ที่ส่งทางไปรษณีย์ ซึ่งส่วนใหญ่จะมีพื้นที่ให้พิมพ์ข้อความต่าง ๆ ไม่มากนัก ดังนั้นในการสร้างหรือพิมพ์บาร์โค้ดแบบ Postnet จึงไม่นิยมสร้างเป็นแถบบาร์โค้ดขนาดใหญ่หรือกว้างเกินไป ด้วยเหตุนี้เองทำให้บาร์โค้ดแบบ Postnet จึงเป็นบาร์โค้ดที่มีขนาดเล็กมากชนิดหนึ่ง

การพิมพ์แถบเส้นบาร์โค้ดของบาร์โค้ดแบบ Postnet นั้น นอกจากเราต้องพิมพ์ส่วนที่เป็นแท่งบาร์โค้ดแล้ว ในการพิมพ์แท่งทึบแต่ละแท่ง เราจะต้องใส่ช่องว่างระหว่างแท่งบาร์โค้ดแต่ละแท่งลงไปด้วย ในรูปที่ 5 จะเป็นตัวอย่างการหารูปแบบการพิมพ์แถบเส้นบาร์โค้ดสำหรับเครื่องพิมพ์ดอตเมตริกซ์ แบบ 24 เข็มพิมพ์ โดยในรูปที่ 5 นี้จะมีการพิมพ์แถบเส้นบาร์โค้ดที่มีความหนาเท่ากับ 1 เส้นพิมพ์ ดังนั้นค่าที่จะส่ง

ออกไปพิมพ์ยังเครื่องพิมพ์สำหรับแท่งสี่เหลี่ยมแบบยาวคือ 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00 ส่วนค่าที่จะส่งไปยังเครื่องพิมพ์สำหรับแท่งสี่เหลี่ยมแบบสั้นจะเป็น 0x00, 0x07, 0xFF, 0x00, 0x00, 0x00 ตามลำดับ สำหรับค่า 0x00 ที่ต่อท้ายนั้นจะเป็นการสั่งให้เครื่องพิมพ์พิมพ์ช่องว่างเพิ่มขึ้นอีกหนึ่งเส้น

ตัวอย่างโปรแกรมพิมพ์บาร์โค้ดแบบ Postnet

สำหรับตัวอย่างของโปรแกรมที่ใช้พิมพ์บาร์โค้ดที่ผมนำมาเสนอในครั้งนี้ จะเป็นโปรแกรมที่สร้างด้วยภาษาซี โดยโปรแกรมจะทำการพิมพ์แถบเส้นบาร์โค้ดที่ใช้แทนรหัสไปรษณีย์ "10240" ทางเครื่องพิมพ์ดอตเมตริกซ์ Epson แบบ 24 เข็มพิมพ์ โดยรายละเอียดของโปรแกรมชุดนี้มีดังนี้

คำสั่งของเครื่องพิมพ์ของ Epson ที่ใช้สำหรับพิมพ์แถบเส้นบาร์โค้ด คือ

ESC * m n1 n2 data

ซึ่งเป็นการบอกเครื่องพิมพ์ให้พิมพ์ในโหมดกราฟิก
โดยที่ m เป็นระดับความละเอียดของการพิมพ์
ในที่นี้มีค่าเท่ากับ 39 ซึ่งหมายถึงเลือกพิมพ์ระดับ

Triple-density

n1, n2 หมายถึงจำนวนคอลัมน์ที่จะพิมพ์ โดยนำ 256
หารจำนวนคอลัมน์ที่จะพิมพ์ทั้งหมดใน 1 ชุด
ผลที่ได้ให้ใส่แทน n2 เศษที่เหลือจะใส่แทนที่
n1
data หมายถึงข้อมูลที่จะส่งไปพิมพ์ 1 ชุด

บรรทัดที่ 013 ถึง 018 เป็นการกำหนดรูปแบบของแถบเส้นบาร์โค้ดที่จะพิมพ์ออกทางเครื่องพิมพ์ โดยจะประกอบด้วยแถบเส้นบาร์โค้ดแบบยาวและแถบเส้นบาร์โค้ดแบบสั้นตามลำดับ โดยแถบเส้นบาร์โค้ดที่พิมพ์จากโปรแกรมชุดนี้จะมีความหนาเท่ากับ 3 เส้นพิมพ์ ส่วนช่องว่างก็จะมีขนาด 3 เส้นพิมพ์เช่นเดียวกัน บรรทัดที่ 022 ถึง 039 เป็นฟังก์ชันสำหรับคำนวณหาค่าตรวจสอบความถูกต้องของรหัสหรือค่า CHECK DIGIT ของรหัสไปรษณีย์ที่จะให้โปรแกรมชุดนี้พิมพ์ โดยในตัวอย่างค่าตรวจสอบความถูกต้องของรหัสจะมีค่าเท่ากับ 3

บรรทัดที่ 040 ถึง 050 เป็นฟังก์ชันที่ทำหน้าที่พิมพ์แถบเส้นบาร์โค้ดแบบยาว

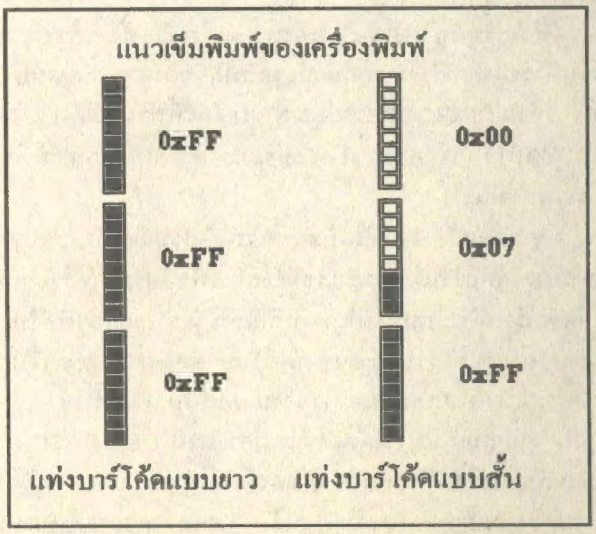
บรรทัดที่ 051 ถึง 061 เป็นฟังก์ชันที่ทำหน้าที่พิมพ์แถบเส้นบาร์โค้ดแบบสั้น

บรรทัดที่ 062 ถึง 153 เป็นฟังก์ชันที่ทำหน้าที่สั่งพิมพ์แถบเส้นบาร์โค้ดส่วนที่ใช้แทนค่าตัวเลข โดยภายในฟังก์ชันนี้จะมีการสั่งพิมพ์แถบเส้นบาร์โค้ดตามรูปแบบหรือ Pattern ของบาร์โค้ดที่ใช้แทนค่าตัวเลขแต่ละตัว บรรทัดที่ 154 ถึง 167 เป็นฟังก์ชันที่ทำหน้าที่นำตัวเลขรหัสไปรษณีย์และค่าตรวจสอบความถูกต้องของรหัสมาพิมพ์เป็นแถบเส้นบาร์โค้ด โดยการทำงานของฟังก์ชันชุดนี้จะเริ่มจากนำค่าที่จะพิมพ์บาร์โค้ดไปหาค่าตรวจสอบความถูกต้องของรหัส (CHECK DIGIT) แล้วนำมารวมต่อกับชุดข้อมูลที่จะพิมพ์ จากนั้นโปรแกรมจะพิมพ์แถบเส้นบาร์โค้ดส่วนเริ่มต้น (start bar) ก่อน และพิมพ์แถบเส้นบาร์โค้ดที่แทนตัวเลขทั้งหมด จากนั้นจึงจะพิมพ์แถบเส้นบาร์โค้ดส่วนปิดท้าย (stop bar) ตามลำดับ บรรทัดที่ 168 ถึง 172 เป็นส่วนโปรแกรมหลักของงาน ซึ่งใช้เป็นตัวอย่างในการเรียกฟังก์ชัน สำหรับพิมพ์แถบบาร์โค้ดขึ้นมาทำงาน

ตัวอย่าง การหาค่าเช็คดิจิตของ 10240

1) นำเลขทั้งหมดมาบวกกัน
   1+0+2+4+0 = 7
2) หารด้วย 10 เก็บเฉพาะเศษไว้
   7 / 10 เศษ = 7
3) นำเศษไปลบออกจาก 10 อีกครั้งหนึ่ง
   10 - 7 = 3 ค่าเช็คดิจิตคือ 3

รูปที่ 4 วิธีการหาค่า CHECK DIGIT



รูปที่ 5 ตัวอย่างการหา Pattern สำหรับเครื่องพิมพ์

ตัวอย่างโปรแกรมที่ผมนำมาเสนอชุดนี้ เป็นเพียงตัวอย่างที่แสดงให้เห็นถึงวิธีการพิมพ์บาร์โค้ดชนิดนี้ทางเครื่องพิมพ์เท่านั้น แต่ยังคงจะนำไปใช้งานจริงไม่ได้ ท่านผู้อ่านต้องนำไปปรับปรุงและพัฒนาต่อเพื่อให้การทำงานของโปรแกรมครบถ้วนสมบูรณ์ และสำหรับเรื่องราวของบาร์โค้ดแบบ Postnet ผมคงต้องขอจบเพียงเท่านี้ พบกันใหม่โอกาสหน้าครับ

ลิสติงที่ 1 Postnet.c

```
Line Source File :POSTNET.C

001 : /*-------------------------------------*/
002 : /*PROGRAM : POSTNET.C เป็นโปรแกรมพิมพ์บาร์โค้ดชนิด POSTNET โดย */
003 : /* พิมพ์ค่า 10240 ทางเครื่องพิมพ์ดอตเมตริกซ์ */
004 : /* Epson 24 เข็มพิมพ์ */
005 : /*PROGRAMMER: ธาวินทร์ ธีลาคุณารักษ์ 30 ธันวาคม 2540 */
006 : /*-------------------------------------*/
007 : #include <stdlib.h>
008 : #include <stdio.h>
009 : #include <string.h>
010 : /*-------------------------------------*/
011 : /* รูปแบบแท่งบาร์โค้ด */
012 : /*-------------------------------------*/
013 : char hight_bar[18] = { 0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,
014 : 0xFF,0xFF,0xFF,0x00,0x00,0x00,
015 : 0x00,0x00,0x00,0x00,0x00,0x00 };
016 : char low_bar[18] = { 0x00,0x07,0xFF,0x00,0x07,0xFF,
017 : 0x00,0x07,0xFF,0x00,0x00,0x00,
018 : 0x00,0x00,0x00,0x00,0x00,0x00 };
019 : /*-------------------------------------*/
020 : /* ฟังก์ชันที่เกี่ยวข้องกับการพิมพ์บาร์โค้ด */
021 : /*-------------------------------------*/
022 : int check_digit (char *text) /*ทำหน้าที่หาค่า check digit */
023 : {
024 : char ch[2];
025 : int data1;
026 : int i;
027 :
028 : data1 = 0;
029 : i = 0;
030 : while  (text[i] != '\0') { /*หาผลรวมของตัวเลขทั้งหมด */
031 : ch[0] = text[i];
032 : ch[1] = '\0';
033 : data1 = data1 + atoi (ch);
034 : i++;
035 : }
036 : data1 = 10- (data1 % 10); /*หาตัวเลขที่บวกกับผลแล้วทำให้ */
037 : return (data1); /*ผลรวมทั้งหมดเป็นเลขเต็มสิบ */
038 : }
039 : /*-------------------------------------*/
040 : void print_hight_bar () /*ทำหน้าที่พิมพ์แท่งสี่เหลี่ยมแบบยาว*/
041 : {
042 : int i;
043 :
044 : fprintf (stdprn,"\x1B*%c%c%c",39,6,0);/*เซตเป็นกราฟิกโหมด*/
045 : for  ( i= 0; i<18 ;i++)
046 : {
047 : fprintf (stdprn,"%c",hight_bar[i]);/* พิมพ์เส้นบาร์โค้ด */
048 : }
049 : }
050 : /*-------------------------------------*/
051 : void print_low_bar () /*ทำหน้าที่พิมพ์แท่งสี่เหลี่ยมแบบสั้น */
052 : {
053 : int i;
054 :
055 : fprintf (stdprn,"\x1B*%c%c%c",39,6,0);/*เซตเป็นกราฟิกโหมด*/
056 : for  ( i= 0; i<18 ;i++)
057 : {
```

→

ลิสติงที่ 1 (ต่อ)

```
058 : fprintf (stdprn,"%c",low_bar[i]); /* พิมพ์เส้นบาร์โค้ด */
059 : }
060 : }
061 : /*-------------------------------------*/
062 : void print_bar_digit (char *digit)/* ฟาหน้าที่พิมพ์แถบ BARCODE */
063 : { /* ที่แทนตัวเลขครั้งละ 1 หลัก */
064 : int i;
065 :
066 : i = 0;
067 : while  (digit[i] != '\0') {
068 : switch  (digit[i]) { /*เลือก pattern*/
069 : case '0' : { /* 1,1,0,0,0 */
070 : print_hight_bar ();
071 : print_hight_bar ();
072 : print_low_bar ();
073 : print_low_bar ();
074 : print_low_bar ();
075 : break;
076 : }
077 : case '1' : { /* 0,0,0,1,1 */
078 : print_low_bar ();
079 : print_low_bar ();
080 : print_low_bar ();
081 : print_hight_bar ();
082 : print_hight_bar ();
083 : break;
084 : }
085 : case '2' : { /* 0,0,1,0,1 */
086 : print_low_bar ();
087 : print_low_bar ();
088 : print_hight_bar ();
089 : print_low_bar ();
090 : print_hight_bar ();
091 : break;
092 : }
093 : case '3' : { /* 0,0,1,1,0 */
094 : print_low_bar ();
095 : print_low_bar ();
096 : print_hight_bar ();
097 : print_hight_bar ();
098 : print_low_bar ();
099 : break;
100 : }
101 : case '4' : { /* 0,1,0,0,1 */
102 : print_low_bar ();
103 : print_hight_bar ();
104 : print_low_bar ();
105 : print_low_bar ();
106 : print_hight_bar ();
107 : break;
108 : }
109 : case '5' : { /* 0,1,0,1,0 */
110 : print_low_bar ();
111 : print_hight_bar ();
112 : print_low_bar ();
113 : print_hight_bar ();
114 : print_low_bar ();
115 : break;
116 : }
117 : case '6' : { /* 0,1,1,0,0 */
118 : print_low_bar ();
119 : print_hight_bar ();
120 : print_hight_bar ();
121 : print_low_bar ();
122 : print_low_bar ();
123 : break;
124 : }
125 : case '7' : { /* 1,0,0,0,1 */
126 : print_hight_bar ();
127 : print_low_bar ();
128 : print_low_bar ();
129 : print_low_bar ();
130 : print_hight_bar ();
```

→

ลิสติงที่ 1 (ต่อ)

```
131 : break;
132 : }
133 : case '8' : { /* 1,0,0,1,0 */
134 : print_hight_bar ();
135 : print_low_bar ();
136 : print_low_bar ();
137 : print_hight_bar ();
138 : print_low_bar ();
139 : break;
140 : }
141 : case '9' : { /* 1,0,1,0,0 */
142 : print_hight_bar ();
143 : print_low_bar ();
144 : print_hight_bar ();
145 : print_low_bar ();
146 : print_low_bar ();
147 : break;
148 : }
149 : }
150 : i++;
151 : }
152 : }
153 : /*-------------------------------------*/
154 : void postnet (char *text) /* ทำหน้าที่นำค่าจาก text มาพิมพ์*/
155 : { /* เป็นแถบเส้น barcode */
156 : int j;
157 : char digit[1];
158 :
159 : itoa (check_digit (text),digit,10);/*หาค่า check digit */
160 : strcat (text,digit);
161 : fprintf (stdprn," ");
162 : print_hight_bar (); /*พิมพ์ส่วน start bar*/
163 : print_bar_digit (text);
164 : print_hight_bar (); /*พิมพ์ส่วน stop bar */
165 : fprintf (stdprn,"\n");
166 : }
167 : /*-------------------------------------*/
168 : void main ()
169 : {
170 : postnet ("10240");
171 : }
172 : /*-------------------------------------*/
```